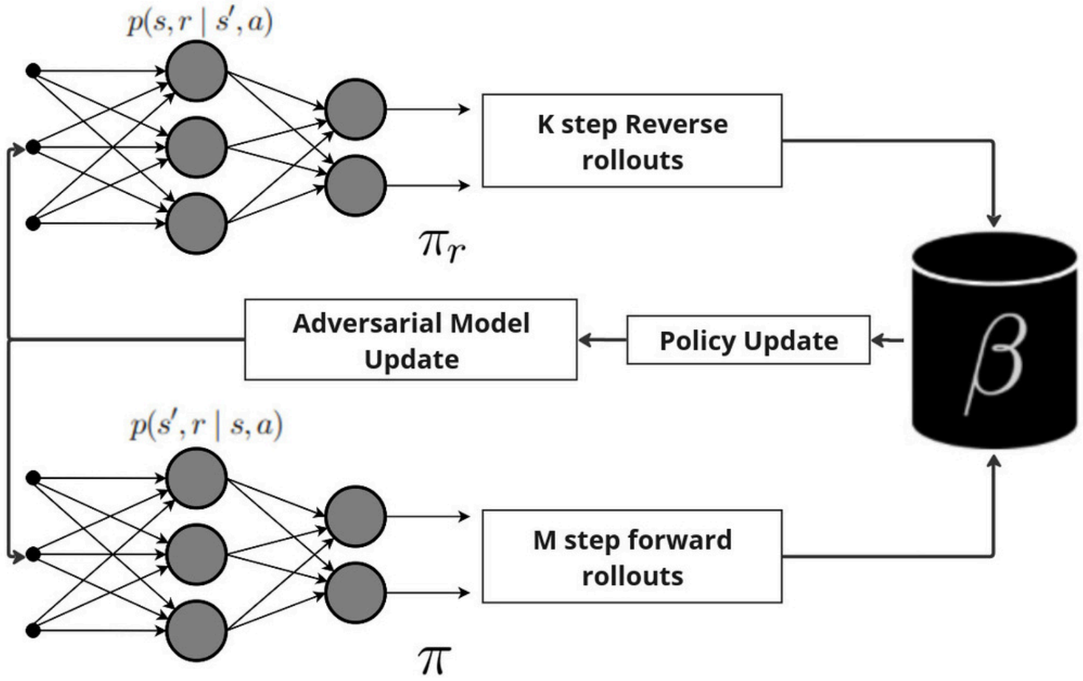
DA7400 Project Endterm Review

Adversarial Offline RL with Bidirectional Imagination

Navin Sriram ED21B044 | Anirud N CE21B014

Problem Statement Recap

Analysing effects of incorporating bidirectional **imagination,** using a **reverse and a forward dynamics model** to generate **reverse and forward rollouts** for **dataset augmentation** while adversarially optimizing the state dynamics with a policy-gradient style approach. We were originally motivated by two papers and their future work sections, RAMBO-RL (Robust adversarial Model Based Offline RL) and ROMI (Offline RL with Reverse Model Imagination)



Dataset Augmentation

- While offline RL has successfully learned complex control policies, it typically requires **large amounts of expert-quality data** to learn effective policies that **generalize to out-of-distribution** states.

- Dataset augmentation with conservative generalization boosts offline RL by enriching suboptimal datasets. In **model-based augmentation**, a learned environment model generates synthetic transitions or trajectories to enhance data diversity and coverage.

- This allows agents to learn from unseen scenarios and improve generalization across different scenarios. Model-based dataset Augmentation methods like **RAMBO-RL** and **ROMI** outperform most offline RL benchmarks

Reverse Imagination

- **Reverse imaginations** (RI) generate possible traces leading to **target goal states inside the offline dataset**, providing a conservative way to augment the offline dataset.

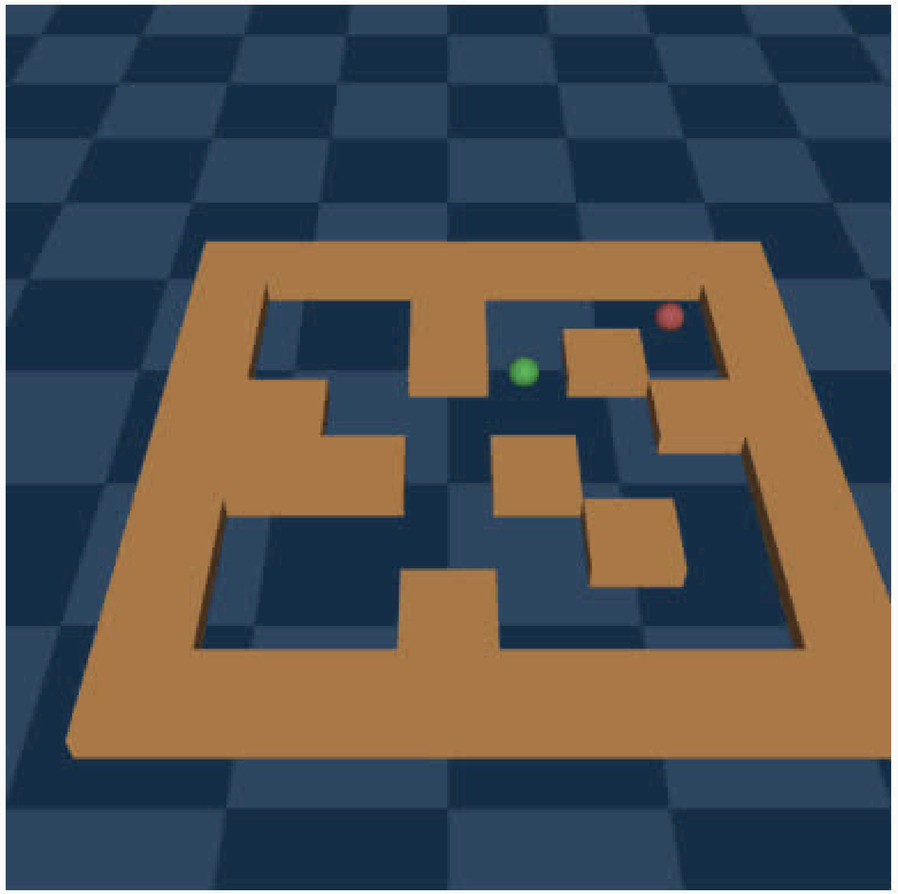
- By simulating reverse trajectories, the model can generate experiences that adhere closely to the high-confidence areas of the original dataset.

- Recent Works like ROMI show the success of such methods, but a key assumption involved is a **near-expert dataset** to build the reverse dynamics model. ROMI trains this in a single shot, rolls out trajectories and uses model-free offline RL methods after this.

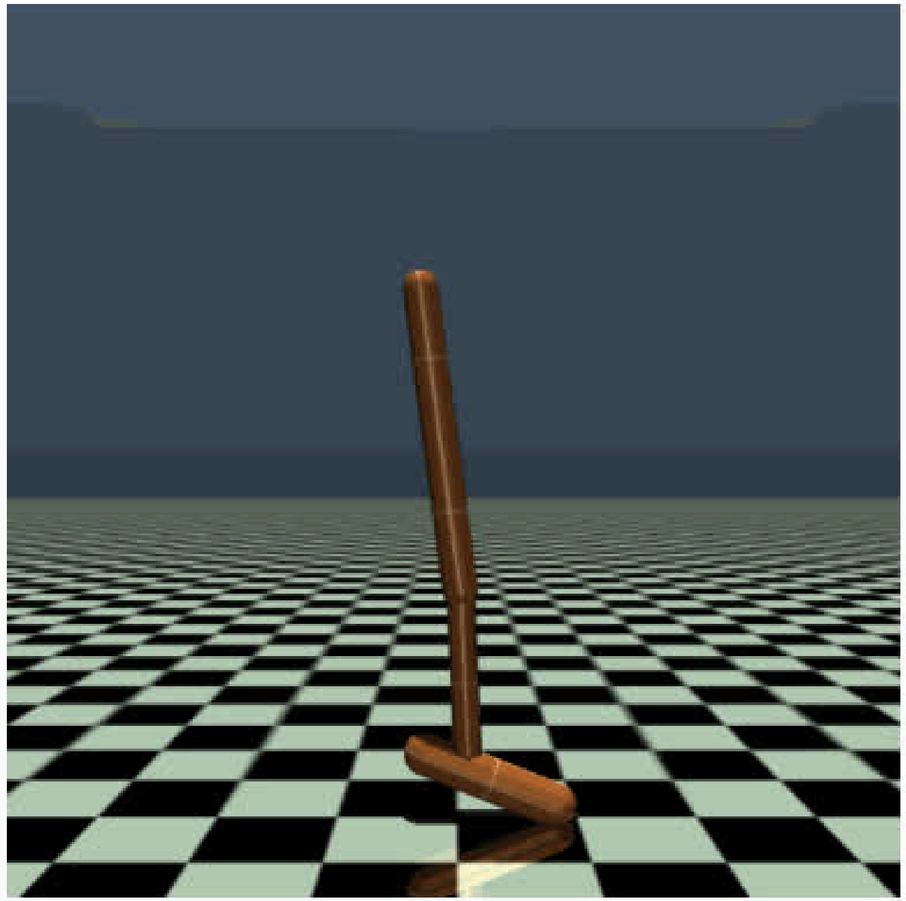
Proposed ?

- We further **update the model adversarially** with the policy to see if it makes it robust to noisy demonstrations. Additionally, we **reinforce RI with Forward Imagination** which is one of the stated future works of the authors

Visualisations of Reverse Policy



Maze2D Medium



Hopper Medium

Adversarial Dynamic Updates

- Efficiently handles the problem of **distributional shift** using adversarial dynamic model updates when used with Forward Dynamics Models

- Adversarially modifies the transition dynamics to enforce conservatism. RAMBO-RL introduced this as a **Two-player 0-sum game** where the dynamics tries to minimize the value function, and the policy tries to maximize it

- Initially optimistic and gradually pessimistic - the gradual introduction of pessimism prevents getting stuck in local maxima

Proposed Changes

- We propose a two simultaneous 2**-player 0-sum adversarial game** between the policy, forward and reverse dynamics model. More conservatism is needed, especially for AntMaze environments, where current Model-based approaches fail.

Methodology

Algorithm 1 Forward and Reverse Rollout with Adversarial Updates

Require: Normalized dataset $\mathcal{D}$

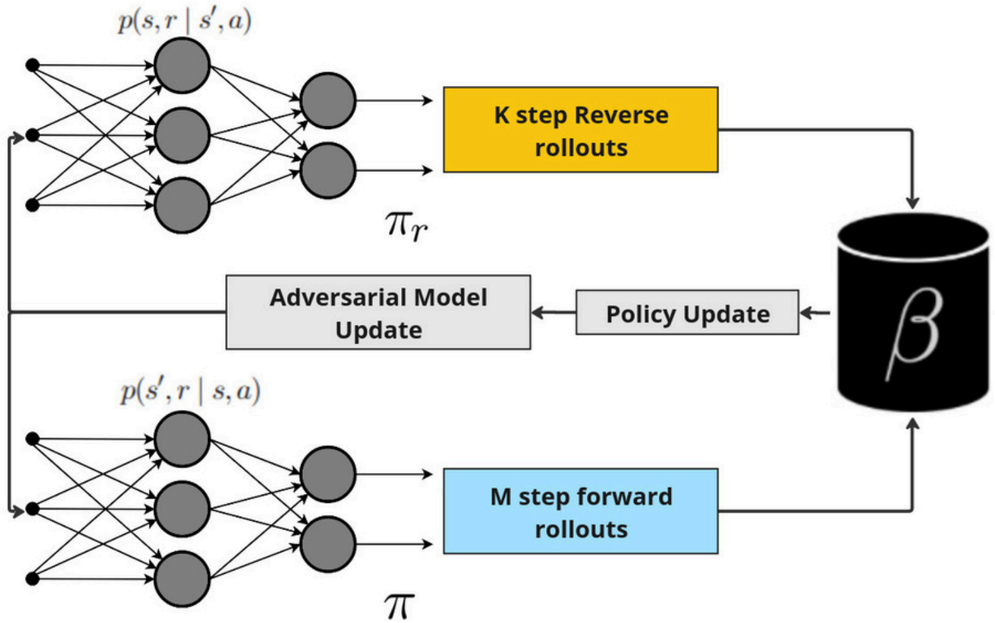
1: $F_{\phi_1} \leftarrow$ Forward dynamics model - MLE
2: $R_{\phi_2} \leftarrow$ Reverse dynamics model - MLE
3: $\pi_f \leftarrow$ Forward Rollout Policy
4: $\pi_r \leftarrow$ Reverse Rollout Policy
5: **for** $i = 1, 2, \ldots, n_{\text{iter}}$ **do**
6: Generate k-step forward rollouts using π_f Add transition data to $\mathcal{D}_{F_{\phi_1}}$
7: Generate k-step reverse rollouts using π_r. Add transition data to $\mathcal{D}_{R_{\phi_2}}$
8: **Agent update:** Update π_f and π_r using samples from $\mathcal{D} \cup \mathcal{D}_{F_{\phi_1}} \cup \mathcal{D}_{R_{\phi_2}}$ using a SAC Framework
9: **Adversarial updates:** Adversially update the forward and reverse transition dynamics models F_{ϕ_1} and R_{ϕ_2}
10: **end for**

$$\mathcal{L}_\phi = \lambda V_\phi^\pi - \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\log \widehat{T}_\phi(s', r | s, a) \right]$$

With a probability of f sample from real data and 1-f from synthetic data

Algorithm specifics	RAMBO	ROMI	OUR approach
Adversarial Updates	Yes	No	Yes
Reverse model imagination	No	Yes	Yes
Multi-agent game	2	-	2 - 2

Methodology



$$\mu, \log \sigma^2 = \text{model}()$$
$$\sigma^{-2} = \exp(-\log \sigma^2)$$
$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left((y_{ij} - \mu_{ij})^2 \cdot \sigma_{ij}^{-2} \right)$$
$$+ \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \log \sigma_{ij}^2 + D$$
$$\mathcal{L} = \mathcal{L} + \lambda \sum \text{max\_logvar} - \lambda \sum \text{min\_logvar}$$

Dynamics pre-training loss, MLE

policy pre-training loss, ->MSE

Model Gradient

$$d_M^\pi(s) := \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi, M)$$

$$\nabla_\phi V_\phi^\pi = \mathbb{E}_{s\sim d_\phi^\pi, a\sim\pi, (s',r)\sim\widehat{T}_\phi}\left[(r + \gamma V_\phi^\pi(s')) \cdot \nabla_\phi \log \widehat{T}_\phi(s', r|s, a)\right]$$

$$\nabla_\phi V_\phi^\pi = \mathbb{E}_{s\sim d_\phi^\pi, a\sim\pi, (s',r)\sim\widehat{T}_\phi}\left[(r + \gamma V_\phi^\pi(s') - Q_\phi^\pi(s,a)) \cdot \nabla_\phi \log \widehat{T}_\phi(s', r|s, a)\right]$$

Expected Value of taking a state s and following the policy thereafter

Adding a Baseline independent of s' and r and reduces Variance and Helps Learning Advantage terms are normalized

How different is Model Gradient from Policy gradient?
• Updates the Likelihood of successor states in the model, instead of the actions in a policy • Advantage term compares the realized outcome to the expected outcome of the current state-action pair.

Rollout

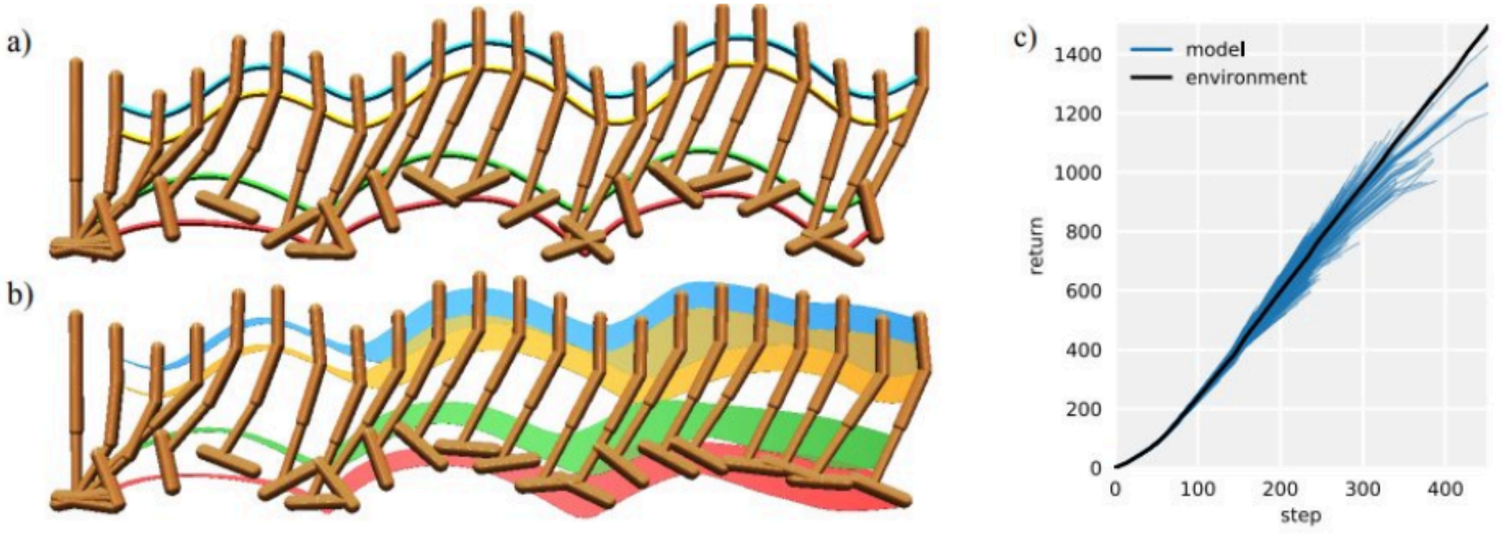


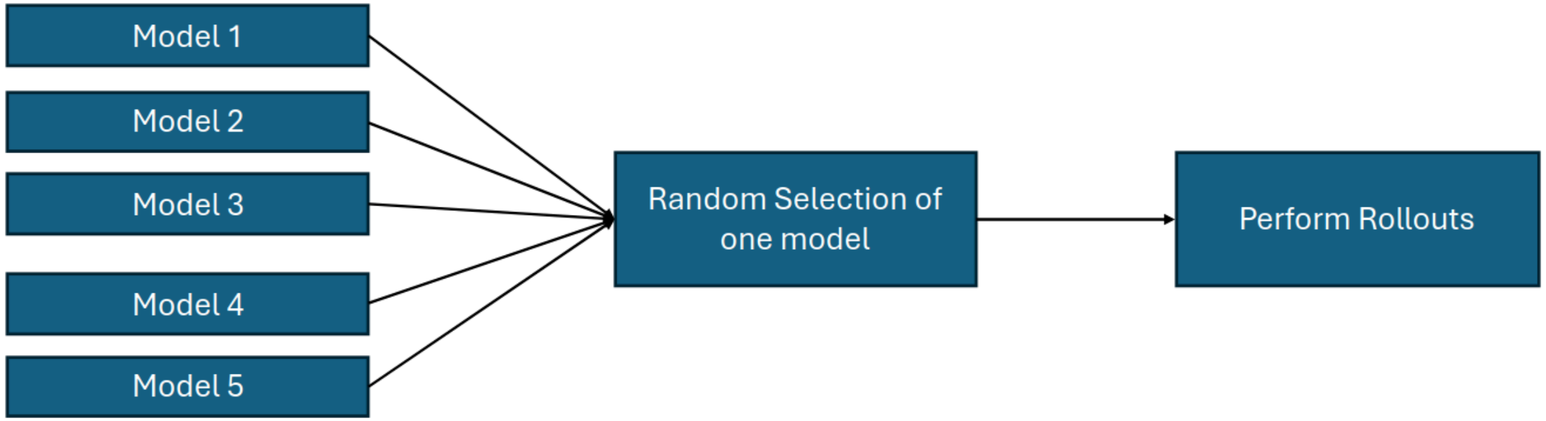
Figure 4: a) A 450-step hopping sequence performed in the real environment, with the trajectory of the body's joints traced through space. b) The same action sequence rolled out under the model 1000 times, with shaded regions corresponding to one standard deviation away from the mean prediction. The growing uncertainty and deterioration of a recognizable sinusoidal motion underscore accumulation of model errors. c) Cumulative returns of the same policy under the model and actual environment dynamics reveal that the policy is not exploiting the learned model. Thin blue lines reflect individual model rollouts and the thick blue line is their mean.

Why k-step Rollouts?

- As the number of steps increases – **Variance** increases
- Need to take only a few steps that are a good representative of the environment

Implementation Details

- Dynamics Model - A **Bayesian Neural Network (BNN)** to model uncertainties in its predictions. We use an **ensemble of 7 networks** and run inference on the **top 5 performers** on the holdout set. We take the MLE loss for the forward and the reverse models.
- **SAC setup for learning the policy**, that learns using the real + fake(fake forward + reverse) replay buffers, where sampling is done with a probability of f from the real buffer and 1-f from the fake buffer.



Chosen based on validated performance on a hold-out set of 1000 transitions in the offline dataset

Key Tunable Hyperparameters

Adversarial loss weight = 3e-4
rollout length = 5
real ratio = 0.5 (ratio of number of transitions sampled from real to synthetic dataset)
Dyn. update frequency = 10000
Actor learning rate 1e-4
Critic learning rate 3e-4

Results so far

	Ours (250)	RAMBO (250)	RAMBO (2000)	ROMI (2000)	COMBO (2000)	MOPO (2000)	MOReL (2000)	CQL	IQL
Hopper Medium	91.42 ± 1.4495	82.68 ± 19.20	96.20 ± 7.116	72.3 ± 17.5	94.9	48	95.4	66.6	66.3
HalfCheetah Med	41.73 ± 2.63	65.61 ± 2.63	78.07 ± 6.61	49.1 ± 0.8	54.2	69.5	42.1	49	47.4

- Were only able to train Hopper and Half Cheetah for about **250 and 150 epochs.**
- Hopper performs almost at par with a fully trained RAMBO and outperforms ROMI
- The standard deviation is comparatively lower in both environments, this could indicate stable learning
- Cheetah performs worse compared to partially trained RAMBO, but shows a gradual increase in the reward

Future work

- Training the implementation for more epochs over different environments
- Hyperparameter tuning and performing an ablation study
- “3 player 0 sum game” - formalizing a competing framework between forward and reverse transition dynamic through adversarial updates
- Perform ablation over the need for adversarial updates

Questions and Suggestions